A PUBLIC KEY CRYPTOGRAPHY BASED ON THE m-INJECTIVITY OF Z_{pqr} OVER ITSELF

[1]Wannarisuk Nongbsap and [2]Dr. Madan Mohan Singh
[1]Assistant Professor, Department of Mathematics, St. Anthony's College-793001, India
[2]Associate Professor, Department of Basic Sciences and Social Sciences,North-Eastern Hill University, Shillong-793022, India

Abstract- *In this paper, we propose a new public key scheme which is a combination of enhanced RSA algorithm [1] and variations of Computational Diffie-Hellman Problem, namely, the Square Computational Diffie-Hellman assumption [2]. Since, the ring Z_{pqr}, where p, q and r are distinct primes, is m-injective over itself so, using one of the ideals of this ring, this public key scheme is proposed, where the primes mentioned are very large.*

Index Terms:- *m-injective, R-monomorphisms, Public Key Cryptography,enhanced RSA algorithm, Square Computational Diffie-Hellman assumption*

1 Introduction

Let R be a ring with identity element. A left ideal I of R is a subgroup of R with respect to addition and for any $x \in I$ and $r \in R$, $rx \in I$. A non-empty set M is a left R-module if M is an abelian group with respect to addition and if there exists a map $. : R \times M \longrightarrow M$ such that (i)$(r+s).m = r.m + s.m$(ii)$r.(m_1 + m_2) = r.m_1 + r.m_2(iii)(rs).m = r.(sm)$ (iv)$1.m = m$, $\forall r, s \in R$ and $\forall m_1, m_2 \in M$[4]. Also, we recall that a function h from a ring R to a ring S is a left R homomorphism if $\forall x, y \in R$, (i)$h(x+y) = h(x) + h(y)$ (ii)$h(rx) = rh(x)$, $\forall r \in R$([4],[5]).

A left R module E is injective over R if for any left R-monomorphism $\alpha : M \to M'$ of left R-modules M and M' and any left R- homomorphism $f : M \to E$, there exists a left R-homomorphism $g : M' \to E$ such that $g_o\alpha = f$[12]. We shall now give Baer's criterion and then modify it to introduce the concept of m-injective rings. Let E be a left R-module. According to Baer's criterion, E is injective over R if and only if for every left ideal A of R, any left R-homomorphism $f : A \longrightarrow E$ can be extended to a left R-homomorphism $g : R \longrightarrow E$([1],[2],[3]). In our study, $E = R$ and using Baer's criterion, we define m-injective in the following manner. A ring R, regarded as a module over itself, is left m-injective over itself if for any left ideal A of R, any left R-monomorphism $f : A \longrightarrow R$ can be extended to a left R-monomorphism $g : R \longrightarrow R$. As in the definition of homomorphism, a function h from a ring R to another ring S is a left R-monomorphism if $\forall x, y \in R$, (i)$h(x+y) = h(x) + h(y)$ (ii)$h(rx) = rh(x)$, $\forall r \in R$ (iii)h is one-one. This concept of rings which are m-injective over themselves is an extension of the concept of Self injective rings which was introduced by Y. Utumi [6] in the year 1965. In [6], Utumi studied the properties of commutative rings which are injective over themselves. In this paper, $R = Z_{pqr}$ is a finite commutative ring.

Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related keys - a public key and a private key. These are non-identical keys and each key performs a unique function. One of the problems used in this paper is enhanced RSA cryptosystem, based on factoring of a number n which is a product of three distinct primes. It stands on the idea that

for a known value e relatively prime to $\phi(n)$(Euler's totient function), there exists inverse d of e modulo $\phi(n)$. Computing $\phi(n)$ is difficult without the knowledge of the prime factors of n and thus d is not easily revealed. The second problem used here is the square Computational Diffie-Hellman Problem i.e given t^d modulo n, it is difficult to find t^{d^2} modulo n, unless d is revealed.
Notation:-$\bar{x}$ used in this paper will denote the integer x modulo n and $\langle\bar{x}\rangle$ will denote the ideal generated by $\bar{x}$.

2 Results and Discussion

Before proposing our public key scheme, we would like to prove the following results based on the m-injectivity of Z_{pqr}.

Theorem 2.1. *Suppose $n = pqr$ then Z_n is m-injective over itself.*

To prove the above theorem, we recall the following results[7].

Proposition 2.2. *For $n \geq 2$, if $\langle\overline{x}_1\rangle = \langle\overline{x}_2\rangle = \langle\overline{x}_3\rangle = ... = \langle\overline{x}_t\rangle$ in Z_n then*
(i)$gcd(x_1, n) = gcd(x_2, n) = gcd(x_3, n) = ... = gcd(x_t, n) = a(say)$
(ii)$\langle\overline{a}\rangle = \langle\overline{x}_1\rangle = \langle\overline{x}_2\rangle = ... = \langle\overline{x}_t\rangle$, where $x_1, x_2, ..., x_t \in \{1, 2, 3, ..., n-1\}$ and $a|n$

Lemma 2.3. *Let $A = \langle\overline{a}\rangle$, as taken above, be an ideal of Z_n and let $f : A \longrightarrow Z_n$, $n \geq 2$, be an R-monomorphism. Let $f(\overline{a}) = \overline{b}$ for some $\overline{b} \in Z_n, \overline{b} \neq \overline{0}$ then*
(i)$a|b$
(ii)If $b|n$ then $b = a$
(iii)$gcd(\frac{n}{a}, \frac{b}{a}) = 1$
(iv)If a is prime then $gcd(n, \frac{n}{a} + \frac{b}{a}) = 1$ or a
(v)If a is prime such that $gcd(n, \frac{n}{a} + \frac{b}{a}) = a$ then $gcd(n, \frac{b}{a}) = 1$.

We shall again prove the following results in order to prove the above theorem.

Lemma 2.4. *Let $n = pqr$ and let $A = \langle\overline{a}\rangle$, as taken above, be a proper ideal of Z_n such that a is a product of any two of the distinct primes p, q and r, say $a = pq$. Let $f : A \longrightarrow Z_n$ be an R-monomorphism. Suppose $f(\overline{a}) = \overline{b}$ for some $\overline{b} \in Z_n$, $\overline{b} \neq \overline{0}$ then*
(i) $gcd(n, \frac{n}{a} + \frac{b}{a}) = 1$ or p or q or a.
(ii)If $gcd(n, \frac{n}{a} + \frac{b}{a}) = p$ then $gcd(n, \frac{pn}{a} + \frac{b}{a}) = 1$ or $gcd(n, \frac{b}{a}) = 1$.
(iii)If $gcd(n, \frac{n}{a} + \frac{b}{a}) = q$ then $gcd(n, \frac{qn}{a} + \frac{b}{a}) = 1$ or $gcd(n, \frac{b}{a}) = 1$.
(iv)If $gcd(n, \frac{n}{a} + \frac{b}{a}) = a$ then $gcd(n, \frac{b}{a}) = 1$

Proof. (i)Let $g = gcd(n, \frac{n}{a} + \frac{b}{a})$ then $g|n$ and $g|\frac{n}{a} + \frac{b}{a}$. This implies that $g|n + b$ showing that $g|b$. By (iii), in Lemma 0.3, we have $g|a$, therefore, $g = 1$ or $g = p$ or $g = q$ or $g = a$.
(ii)Suppose $gcd(n, \frac{n}{a} + \frac{b}{a}) = p$. Let $g = gcd(n, \frac{pn}{a} + \frac{b}{a})$ then $g|n$ and $g|\frac{pn}{a} + \frac{b}{a}$. This implies that $g|pn + b$

showing that $g|b$. By (iii) of Lemma 0.3, we have $g|a$, therefore, $g = 1$ or $g = p$ or $g = q$ or $g = a$.
Again, $g|pqr$ and $g|\frac{pn}{a} + \frac{b}{a} \Rightarrow g|pr + \frac{b}{a} \Rightarrow g|pqr + \frac{b}{a}q$. Now, $p|\frac{n}{a} + \frac{b}{a} \Rightarrow p|r + \frac{b}{a} \Rightarrow r + \frac{b}{a} = pl \Rightarrow \frac{b}{a} = pl - r$.
Therefore, $g|(pl - r)q \Rightarrow g|plq - rq$.
If $g = p$ then $p|rq$, a contradiction.
If $g = a$ then $a|rq \Rightarrow p|r$, a contradiction. Hence, $g = 1$ or $g = q$.
Suppose $g = q$ i.e, $gcd(n, \frac{pn}{a} + \frac{b}{a}) = q$. Let $g' = gcd(n, \frac{b}{a})$ then $g'|gcd(n, b)$ which implies that $g'|a$.
Therefore $g' = 1$ or p or q or a
If $g' = p$ then $p|\frac{b}{a}$, therefore, $p|\frac{pn}{a} + \frac{b}{a}$. So, $p|gcd(n, \frac{pn}{a} + \frac{b}{a})$, which implies that $p|q$, a contradiction.
If $g' = q$ then $q|\frac{b}{a}$. This implies that $q|p\frac{n}{a}$ and therefore, $q|pr$, a contradiction.
if $g' = a$ then $a^2|b \Rightarrow b = a^2k$, for some positive integer k.
Now, $q = gcd(n, \frac{pn}{a} + \frac{b}{a}) = gcd(pqr, pr + ak) = gcd(pqr, pr + pqk) = pgcd(qr, r + qk)$. This implies that $p|q$, a contradiction. Hence $g' = 1$.
(iii)proof same as (ii).
(iv)Let $g = gcd(n, \frac{b}{a})$ hence $g|a$ so by the given condition, $g|(\frac{n}{a} + \frac{b}{a})$ and therefore, $g|\frac{n}{a}$. This implies that $g|gcd(\frac{n}{a}, \frac{b}{a}) \Rightarrow g|1 \Rightarrow g = 1$.
The proof will be similar if $a = qr$ or $a = pr$. □

With the above results, we are now in a position to prove the above theorem.

Proof. Let A be an ideal of Z_n and let $f : A \longrightarrow Z_n$ be a left R-monomorphism. Suppose $A = 0$ then taking $g : Z_n \longrightarrow Z_n$ to be the identity map, we find that g is a left R- monomorphism extending f.
Suppose $A = Z_n$ then taking $g : Z_n \longrightarrow Z_n$ to be equal to f, we find that g is a left R-monomorphism extending f. Suppose A is a nonzero proper ideal of Z_n. Then $A = \langle \overline{a} \rangle$ where $\overline{a} \neq \overline{0}$ and $a = p$ or $a = q$ or $a = r$ or $a = pq$ or $a = qr$ or $a = pr$. Let $f : A \longrightarrow Z_n$ be defined by $f(\overline{a}) = \overline{b}$ where $\overline{b} \neq \overline{0}$.
If $a = p$ or q or r,
We define $g : Z_n \longrightarrow Z_n$ by

$$g(\overline{x}) = \begin{cases} \overline{\left(\frac{n}{a} + \frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{n}{a} + \frac{b}{a}) = 1 \\ \overline{\left(\frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{b}{a}) = 1 \end{cases}$$

Clearly, g is a well-defined left R homomorphism which extends f. For the first case, if $\overline{x} \in Kerg \Rightarrow g(\overline{x}) = \overline{0} \Rightarrow \overline{(\frac{n}{a} + \frac{b}{a})}\overline{x} = \overline{0} \Rightarrow n|(\frac{n}{a} + \frac{b}{a})x$.
Since $(n, \frac{n}{a} + \frac{b}{a}) = 1 \Rightarrow n|x \Rightarrow \overline{x} = \overline{0}$ showing that g is one-one. Similarly, we can prove for the second case also. Therefore, g is a left R-monomorphism.(By [7])
Suppose $a = pq$. We define $g : Z_n \longrightarrow Z_n$ by

$$g(\overline{x}) = \begin{cases} \overline{\left(\frac{n}{a}+\frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{n}{a}+\frac{b}{a}) = 1 \\ \overline{\left(\frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{b}{a}) = 1 \\ \overline{\left(\frac{pn}{a}+\frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{pn}{a}+\frac{b}{a}) = 1 \\ \overline{\left(\frac{qn}{a}+\frac{b}{a}\right)}\overline{x}, & \text{if } (n, \frac{qn}{a}+\frac{b}{a}) = 1 \end{cases}$$

.

It is easy to see that g is a left R-monomorphism extending f.
Similarly, for $a = qr$ and $a = pr$, we can find left R-monomorphisms extending f. Hence, the theorem. □

It is to be noted that since g is a one-one function from a finite set to itself, it will be bijective and hence, it will have an inverse.

Before proposing our scheme, we would like to state the following result which is due to the Chinese Remainder Theorem[11].

Proposition 2.5. *Let $gcd(a, b) = 1$ and $c > 0$ then there exists an integer x such that $gcd(a + bx, c) = 1$*

The above result is used in the proposed scheme, in step 7.

The Proposed Public Key Scheme

A user X who wants to create public and private keys has to do the following steps:-
1. Choose three large and distinct primes p, q and r.
2. Compute $n = pqr$.
3. Compute $\phi(n) = (p-1)(q-1)(r-1)$.
4. Compute $a = pq$ and consider the left ideal generated by $\bar{a}$.
5. Choose a monomorphic image $\bar{b}$ of $\bar{a}$, where $b \in \{2, 3, 4, ..., n-1\}$ such that $gcd(n, \frac{n}{a} + \frac{b}{a}) = p$ and $aq \nmid pn + b$.
6. Compute $t = \frac{pn+b}{a}$.
7. Choose $z > 1$ such that $gcd(n + tz, \phi(n)) = 1$ and let $e = n + tz$.
8. Compute d such that $ed \equiv 1(mod\ \phi(n))$.
9. Compute $h \equiv t^d(mod\ n)$.
10. Compute $l \equiv (t^{-1})^{d^2}(mod\ n)$.

The public keys of X are (n, e, h) and the private keys of X are (d, l).

Encryption

The plaintext space is Z_n. Suppose another user Y wants to send a message $\overline{m} \in Z_n$ to X using $X's$ public keys then Y will have to compute $c = hm^e(mod\ n)$.
Y sends to X the encrypted message c.

Decryption

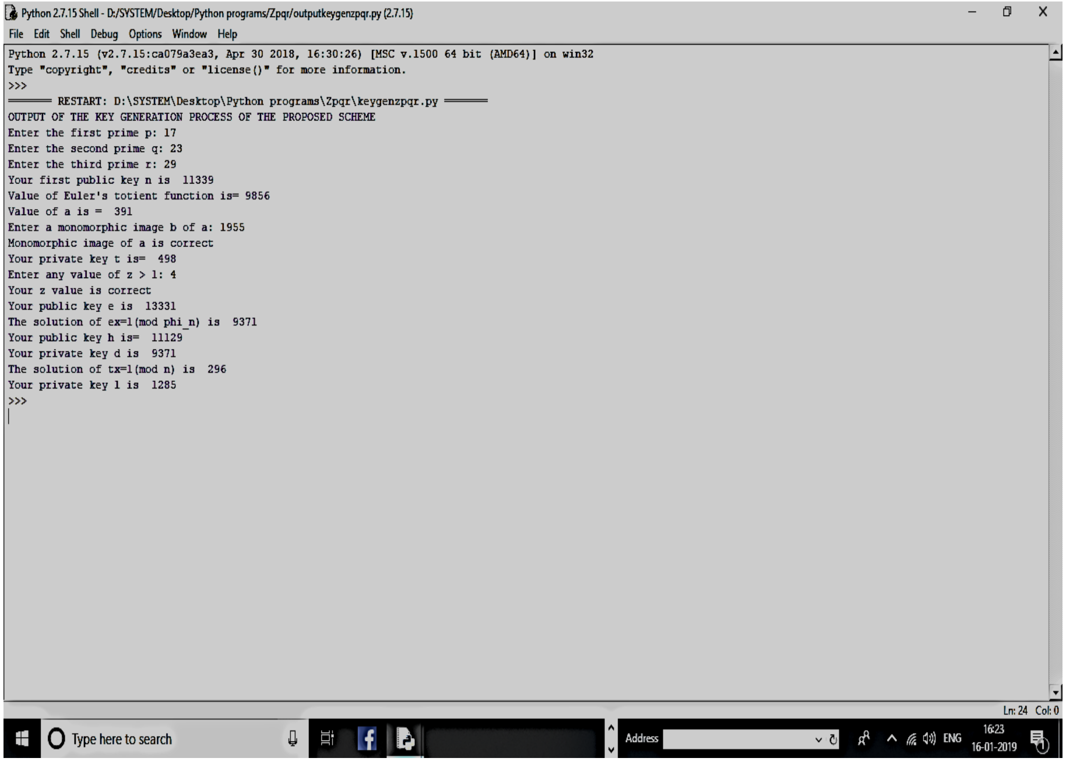
For the decryption of the message c, X should compute $m \equiv c^d l(mod\ n)$, using private keys d and l.
To see how the scheme is implemented, we take an example below.

Example 2.6. *We take $p = 17$, $q = 23$ and $r = 29$. Then $n = pqr = 11339$, $\phi(n) = 9856$, $a = pq = 391$. Choose $b = 1955$ then $\bar{b}$ will be a monomorphic image of $\bar{a}$, where the domain of definition is $\langle\bar{a}\rangle$. Now,*

$gcd(n, \frac{n+b}{a}) = p$, and $aq = 8993$ does not divide $pn + b = 194718$, therefore, $t = \frac{pn+b}{a} = 498$ will be relatively prime to n. Again, choosing $z = 4$, we get $e = n + tz = 13331$, which is relatively prime to $\phi(n)$. Solving $ed \equiv 1(mod\ \phi(n))$, we get $d = 9371$. Computing $h \equiv t^d(mod\ n)$, we get $h = 11129$ modulo n and computing t^{-1} modulo n, we get 296 modulo n. Finally, $l \equiv (t^{-1})^{d^2}(mod\ n)$ gives us $l = 1285$ modulo n. If $m = 456$ modulo n, is the message that we want to send then we encrypt the message through $c = 2134$ modulo n via $c \equiv hm^e(mod\ n)$ using public keys n, e and h. The decryption is carried using private keys d and l to get the message m.

The above processes of key generation, encryption and decryption of messages are shown in the images below. The outputs of the three programs and the elapsed times during the encryption and decryption processes were recorded using Python Language, version 2.7.15, using GNU multi precision library(GMP) on 3.2GHz processor with 4GB RAM.

IMAGE OF THE OUTPUT OF THE KEY GENERATION PROCESS

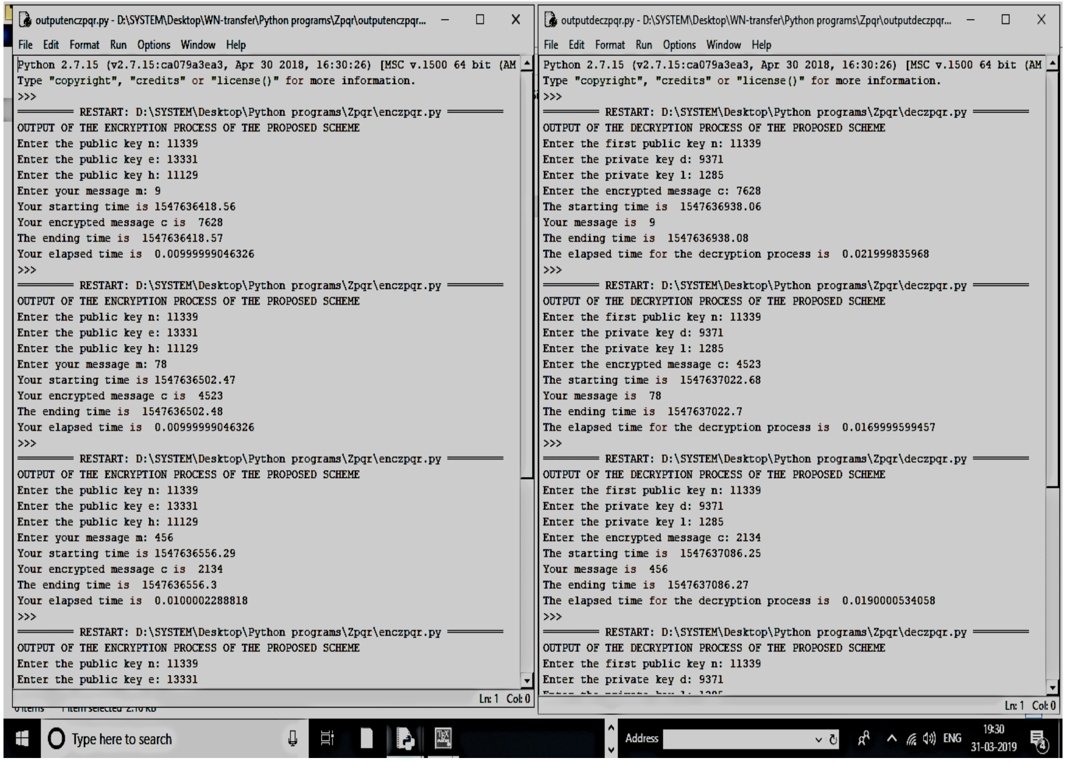


Python 2.7.15 Shell - D:/SYSTEM/Desktop/Python programs/Zpqr/outputkeygenzpqr.py (2.7.15)
File Edit Shell Debug Options Window Help
Ln: 24 Col: 0
Type here to search
Address
ENG
16:23
16-01-2019

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\keygenzpqr.py ======
OUTPUT OF THE KEY GENERATION PROCESS OF THE PROPOSED SCHEME
Enter the first prime p: 17
Enter the second prime q: 23
Enter the third prime r: 29
Your first public key n is  11339
Value of Euler's totient function is= 9856
Value of a is =  391
Enter a monomorphic image b of a: 1955
Monomorphic image of a is correct
Your private key t is=  498
Enter any value of z > 1: 4
Your z value is correct
Your public key e is  13331
The solution of ex=1(mod phi_n) is  9371
Your public key h is=  11129
Your private key d is  9371
The solution of tx=1(mod n) is  296
Your private key l is  1285
>>>
```

fig(i)

IMAGES OF THE OUTPUTS OF THE ENCRYPTION AND DECRYPTION PROCESSES


outputenczpqr.py - D:\SYSTEM\Desktop\WN-transfer\Python programs\Zpqr\outputenczpqr...
File Edit Format Run Options Window Help
Ln: 1 Col: 0
outputdeczpqr.py - D:\SYSTEM\Desktop\WN-transfer\Python programs\Zpqr\outputdeczpqr...
File Edit Format Run Options Window Help
Ln: 1 Col: 0
Type here to search
Address
ENG
19:30
31-03-2019

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AM
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\enczpqr.py ========
OUTPUT OF THE ENCRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the public key n: 11339
Enter the public key e: 13331
Enter the public key h: 11129
Enter your message m: 9
Your starting time is 1547636418.56
Your encrypted message c is  7628
The ending time is  1547636418.57
Your elapsed time is  0.00999999046326
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\enczpqr.py ========
OUTPUT OF THE ENCRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the public key n: 11339
Enter the public key e: 13331
Enter the public key h: 11129
Enter your message m: 78
Your starting time is 1547636502.47
Your encrypted message c is  4523
The ending time is  1547636502.48
Your elapsed time is  0.00999999046326
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\enczpqr.py ========
OUTPUT OF THE ENCRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the public key n: 11339
Enter the public key e: 13331
Enter the public key h: 11129
Enter your message m: 456
Your starting time is 1547636556.29
Your encrypted message c is  2134
The ending time is  1547636556.3
Your elapsed time is  0.0100002288818
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\enczpqr.py ========
OUTPUT OF THE ENCRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the public key n: 11339
Enter the public key e: 13331
```

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AM
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\deczpqr.py ========
OUTPUT OF THE DECRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the first public key n: 11339
Enter the private key d: 9371
Enter the private key l: 1285
Enter the encrypted message c: 7628
The starting time is  1547636938.06
Your message is  9
The ending time is  1547636938.08
The elapsed time for the decryption process is  0.021999835968
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\deczpqr.py ========
OUTPUT OF THE DECRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the first public key n: 11339
Enter the private key d: 9371
Enter the private key l: 1285
Enter the encrypted message c: 4523
The starting time is  1547637022.68
Your message is  78
The ending time is  1547637022.7
The elapsed time for the decryption process is  0.0169999599457
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\deczpqr.py ========
OUTPUT OF THE DECRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the first public key n: 11339
Enter the private key d: 9371
Enter the private key l: 1285
Enter the encrypted message c: 2134
The starting time is  1547637086.25
Your message is  456
The ending time is  1547637086.27
The elapsed time for the decryption process is  0.0190000534058
>>>
======== RESTART: D:\SYSTEM\Desktop\Python programs\Zpqr\deczpqr.py ========
OUTPUT OF THE DECRYPTION PROCESS OF THE PROPOSED SCHEME
Enter the first public key n: 11339
Enter the private key d: 9371
```

fig(ii)

In fig(ii), the first image gives the elapsed times of the encryption processes done while varying the message *m* and in the second image, we have the corresponding elapsed times of the decryption processes. The outputs of the decryption processes have been placed against those of the encryption processes in order to give a clear view of the two tasks that are happening side by side.

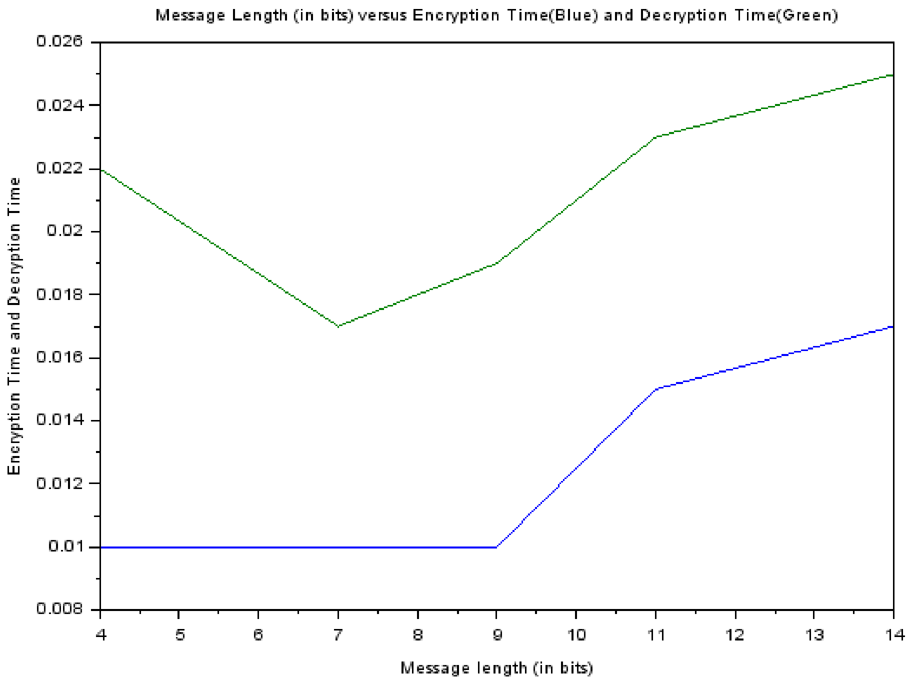
The following table gives us the elapsed time during the encryption and decryption of different messages.

RECORDS OF ELAPSED TIMES

Message(m)	Length(in bits)	Encrypted Message(c)	Encryption time(in ms)	Decryption time(in ms)
9	4	7628	0.00999999046326	0.021999835968
78	7	4523	0.00999999046326	0.0169999599457
456	10	2134	0.0100002288818	0.0190000534058
1892	11	4588	0.0150001049042	0.0230000019073
10783	14	6849	0.0169999599457	0.0250000953674

It can be seen in the table above that the decryption time is slightly higher than the encryption time. The encryption time is consistent at the beginning and then increases. However, the decryption time shows a decrease at the beginning and then eventually increases. This is also highlighted in the graph below. The curve of the decryption time (in green) lies above the curve of the encryption time (in blue). The encryption time curve is parallel to the X-axis at the beginning and then goes up. The decryption time curve slopes down and then moves up. But this varies depending on the values selected in the key generation process.

MESSAGE LENGTH VERSUS ELAPSED TIME GRAPH



fig(iii)

3 Security of the scheme

The security of the scheme is based on integer factorization problem. The above scheme uses three prime factors of n and this increases the security of the scheme. Again, this scheme uses enhanced RSA cryptosystem. An adversary who tries to find the private key d will have to find $\phi(n)$ first, which is an impossible task, for a large n, unless he/she knows the factors of n. Since, the private key l has been computed using d so, this adds to the security of the scheme. According to the Federal Information Processing Standards Publication (FIPS PUBS) the size of n chosen should be a minimum of 1024 or 2048 or 3072 bits. RSA claims that size of n of at least 3072 bits should be used if security is required beyond 2030. Suppose length of n is at least 3072 bits, the size of p, q and r should be at least 1024 bits long. Also, taking a large n can somewhat delay the time of factorizing it via, Polland Rho method or the New Factorization (NF) method[9]. Moreover, larger size of n can also resist brute force attack. Again, p, q and r are to be taken in such a way that they do not permit the applications of known algorithms like the number field sieve method and that they do not give rise to the use of Euclidean Extended algorithm to solve for d in the congruence $ed \equiv 1(mod\ \phi(n))$. Moreover, taking large n can also resist attacks on the Square Computational Diffie-Hellman problem used in this paper. Such attacks are the brute force attack, Shank's Baby-step Giant-step, Pollard's Rho method, Index Calculus Method [10].

An attacker who tries to find m directly will have to compute h^{-1} modulo n. This will lead him/her to the congruence $ch^{-1} \equiv m^e (mod\ n)$, where m is quickly solvable only if he/she knows d. Again, an attacker who tries to find t modulo n can do so using the congruence $h^e \equiv t(mod\ n)$. But again, knowing t and h will not give him/her the value of d via the congruence $h \equiv t^d (mod\ n)$ because this is the discrete logarithm problem which is not easily solvable. Now, suppose the actual value of t is known to the attacker then from the equation $t = \frac{pn}{a} + \frac{b}{a}$, one can get $pr = t - \frac{b}{a}$. This equation contains four unknowns and only one known i.e t. So the factors of n are protected thereby, d is protected. Hence, this scheme is semantically secured and as secured as the other schemes based on enhanced RSA cryptosystem and Square computational Diffie-Hellman Problem.

4 Performance Analysis

The encryption algorithm of the proposed scheme requires one modular exponentiation, viz, m^e and one modular multiplication i.e hm^e. The decryption process requires one modular exponentiation viz, c^d and one modular multiplication, viz, $c^d l$. Hence, the encryption and decryption processes are shorter compared to other existing schemes. One disadvantage of the scheme is that the key generation process is quiet lengthy because of the modular exponentiation $(t^{-1})^{d^2}$.

5 Conclusion and Future Works

In this paper, we used the concept of m-injective Rings to create a Public key Cryptosystem. This property of Z_{pqr} of being m-injective over itself helped in the creation of a public key t which is relatively prime to n. The application of Chinese Remainder Theorem helped in the creation of another public key e which is relatively prime to $\phi(n)$ thus making it as efficient as the enhanced RSA Cryptosystem. Again, the property of t, being relatively prime to n, created another way to the application of the Square Computational Diffie-Hellman problem. So the whole scheme involves around the idea of two hard problems, thus making the proposed scheme as efficient as the other existing schemes.

Through this paper, it can be seen that the theory of rings which are m-injective over themselves have wide applications in the field of cryptography. The property of Z_{pqr} of being m-injective over itself contributed to the creation of this proposed scheme. Hence, our future work will include increasing the prime factors of n and based on the m-injectivity of Z_n, we shall try to propose more schemes based on some other hard problems.

6 References

[1]V. Choudhary, N.Praveen, Enhanced RSA Cryptosystem based on three prime numbers, IJISET-International Journal of Innovative Science, Engineering and Technology, Vol. 1 Issue 10, 753-757, December 2014

[2]F. Bao, R.H. Deng, H.Zhu, Variations of Diffie-Hellman Problem, Variations of Diffie-Hellman Problem, International Conference on Information and Communications Security ICICS 2003: Information and Communications Security, 301-312

[3]R. Baer, Abelian Groups that Are Direct Summands of Every Containing Abelian Group, Bull. Amer. Math. Soc. 46, 800-806, 1940.

[4]C. Faith, Algebra: Rings, Modules and Categories, I. Berlin, p. 157, 1973.

[5]T.Y.Lam, Lectures on Modules and Rings. New York: Springer-Verlag, p. 63, 1999.

[6]Y.Utumi, On Continuous Rings and Self Injective Rings,University of Rochester, Rochester, Newyork (1965)158-173.

[7]W. Nongbsap, M. M. Singh, A Public Key Cryptography Based on the m-injectivity of Z_{pq} over itself, Communicated

[8]P. C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, CRYPTO '96 Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology , 104-113, 1996

[9]B.R.Ambedkar, S.S.Bedi, A New Factorization Method to Factorize RSA Public Key Encryption, IJSCI International Journal of Computer Science Issues, Vol.8, Issue 6 No. 1, November 2011, 242-247

[10]K. Rabah, Security of the Cryptographic Protocols Based on Discrete Logarithm Problem, Journal of Applied Sciences 5(9): 1692-1712, 2005, ISSN 1812-5654.

[11]I.Niven, H.S.Zukerman, H.L.Montgomery, An Introduction to the theory of numbers, Fifth Edition, John Wiley and Sons, Inc., p.73, 2000

[12]M.Kosters, Injective Modules and the Injective Hull of a Module, November 27, 2009